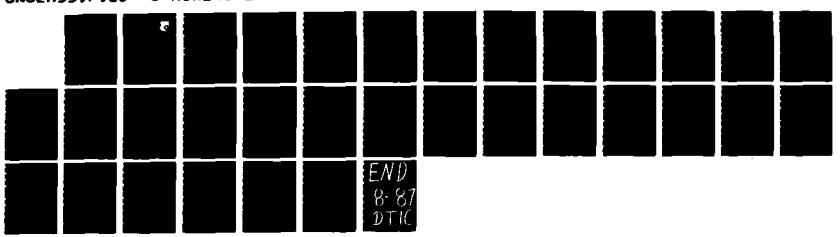
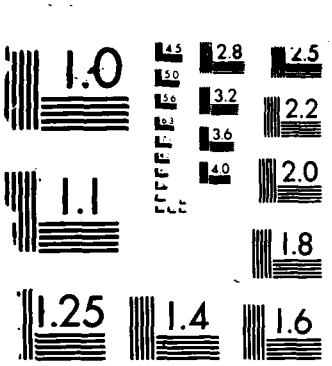
AD-A182 537 INTEGRATED INFORMATION SUPPORT SYSTEM (IISS) VOLUME 8 1/1
USER INTERFACE SUBS (U) GENERAL ELECTRIC CO
SCHENECTADY NY PRODUCTION RESOURCES CONSU
UNCLASSIFIED C MORENC ET AL 01 NOV 85 DS-6201441008 F/G 12/5 NL





AFWAL-TR-86-4006
Volume VIII
Part 2



# INTEGRATED INFORMATION SUPPORT SYSTEM (IISS)
## Volume VIII - User Interface Subsystem
## Part 2 - User Interface Services Development Specification

General Electric Company
Production Resources Consulting
One River Road
Schenectady, New York 12345

Final Report for Period 22 September 1980 - 31 July 1985

November 1985

Approved for public release; distribution is unlimited.

PREPARED FOR:

MATERIALS LABORATORY
AIR FORCE WRIGHT AERONAUTICAL LABORATORIES
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AFB, OH 45433-6533

*NOTICE*

*When Government drawings specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever, and the fact that the government may have formulated furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto*

*This report has been reviewed by the Office of Public Affairs (ASD/PA) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.*

*This technical report has been reviewed and is approved for publication.*

*DAVID L. JUDSON, PROJECT MANAGER*
*AFWAL/MLTC*
*WRIGHT PATTERSON AFB OH 45433*

*5 Aug 1986*
*DATE*

*FOR THE COMMANDER.*

*GERALD C. SHUMAKER, BRANCH CHIEF*
*AFWAL/MLTC*
*WRIGHT PATTERSON AFB OH 45433*

*7 Aug 86*
*DATE*

*"If your address has changed, if you wish to be removed from our mailing list, or if the addressee is no longer employed by your organization please notify AFWAL/MLTC, W-PAFB, OH 45433 to help us maintain a current mailing list."*

*Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.*

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE

1 November 1985

# REPORT DOCUMENTATION PAGE

AD-A182 538

| | |
|---|---|
| 1a. REPORT SECURITY CLASSIFICATION<br>Unclassified | 1b. RESTRICTIVE MARKINGS |
| 2a. SECURITY CLASSIFICATION AUTHORITY | 3. DISTRIBUTION/AVAILABILITY OF REPORT<br>Approved for public release; distribution is unlimited. |
| 2b. DECLASSIFICATION/DOWNGRADING SCHEDULE | |
| 4. PERFORMING ORGANIZATION REPORT NUMBER(S) | 5. MONITORING ORGANIZATION REPORT NUMBER(S)<br>AFWAL-TR-86-4006 Vol VIII, Part 2 |
| 6a. NAME OF PERFORMING ORGANIZATION<br>General Electric Company<br>Production Resources Consulting<br>6b. OFFICE SYMBOL (If applicable) | 7a. NAME OF MONITORING ORGANIZATION<br>AFWAL/MLTC |
| 6c. ADDRESS (City, State and ZIP Code)<br>1 River Road<br>Schenectady, NY 12345 | 7b. ADDRESS (City, State and ZIP Code)<br>WPAFB, OH 45433-6533 |
| 8a. NAME OF FUNDING/SPONSORING ORGANIZATION<br>Materials Laboratory<br>Air Force Systems Command, USAF<br>8b. OFFICE SYMBOL (If applicable)<br>AFWAL/MLTC | 9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER<br>F33615-80-C-5155 |
| 8c. ADDRESS (City, State and ZIP Code)<br>Wright-Patterson AFB, Ohio 45433 | 10. SOURCE OF FUNDING NOS<br>PROGRAM ELEMENT NO. 78011F; PROJECT NO. 7500; TASK NO. 62; WORK UNIT NO. 01 |
| 11. TITLE (Include Security Classification)<br>(See Reverse) | |
| 12. PERSONAL AUTHOR(S) Cross, Valerie and Morenc, Carol and Robie, Penny | |
| 13a. TYPE OF REPORT<br>Final Technical Report | 13b. TIME COVERED<br>22 Sept 1980 - 31 July 1985 |
| 14. DATE OF REPORT (Yr., Mo., Day)<br>1985 November | 15. PAGE COUNT<br>31 |
| 16. SUPPLEMENTARY NOTATION<br>ICAM Project Priority 6201 | The computer software contained herein are theoretical and or references that in no way reflect Air Force-owned or -developed computer software. |

| 17 COSATI CODES | | |
|---|---|---|
| FIELD | GROUP | SUB GR |
| 1308 | 0905 | |
| | | |

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

This specification establishes the development, test, and qualification requirements of a collection of form-based applications identified as the User Interface Services (UIS). Each application is individually accessed by filling its name in the function select form. Each application presents a form to the user which must be filled in with information for processing by that application. The application then communicates with the Form Processor through messages created and sent by the Application Interface and handled by the User Interface Monitor of the Form Processor.

| | |
|---|---|
| 20. DISTRIBUTION/AVAILABILITY OF ABSTRACT<br>UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT ☐ DTIC USERS ☐ | 21. ABSTRACT SECURITY CLASSIFICATION<br>Unclassified |
| 22a. NAME OF RESPONSIBLE INDIVIDUAL<br>David L. Judson | 22b. TELEPHONE NUMBER (Include Area Code)<br>513-255-6976<br>22c. OFFICE SYMBOL<br>AFWAL/MLTC |

DD FORM 1473, 83 APR EDITION OF 1 JAN 73 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE

11. Title

**Integrated Information Support System (IISS)**
**Vol VIII - User Interface Subsystem**
**Part 2 - User Interface Services Development Specification**

## PREFACE

This development specification covers the work performed under Air Force Contract F33615-80-C-5155 (ICAM Project 6201). This contract is sponsored by the Materials Laboratory, Air Force Systems Command, Wright-Patterson Air Force Base, Ohio. It was administered under the technical direction of Mr. Gerald C. Shumaker, ICAM Program Manager, Manufacturing Technology Division, through Project Manager, Mr. David Judson. The Prime Contractor was Production Resources Consulting of the General Electric Company, Schenectady, New York, under the direction of Mr. Alan Rubenstein. The General Electric Project Manager was Mr. Myron Hurlbut of Industrial Automation Systems Department, Albany, New York.

Certain work aimed at improving Test Bed Technology has been performed by other contracts with Project 6201 performing integrating functions. This work consisted of enhancements to Test Bed software and establishment and operation of Test Bed hardware and communications for developers and other users. Documentation relating to the Test Bed from all of these contractors and projects have been integrated under Project 6201 for publication and treatment as an integrated set of documents. The particular contributors to each document are noted on the Report Documentation Page (DD1473). A listing and description of the entire project documentation system and how they are related is contained in document FTR620100001, Project Overview.

The subcontractors and their contributing activities were as follows:

TASK 4.2

| Subcontractors | Role |
|---|---|
| Boeing Military Aircraft Company (BMAC) | Reviewer. |
| D. Appleton Company (DACOM) | Responsible for IDEF support, state-of-the-art literature search. |
| General Dynamics/ Ft. Worth | Responsible for factory view function and information models. |

| Subcontractors | Role |
|---|---|
| Illinois Institute of Technology | Responsible for factory view function research (IITRI) and information models of small and medium-size business. |
| North American Rockwell | Reviewer. |
| Northrop Corporation | Responsible for factory view function and information models. |
| Pritsker and Associates | Responsible for IDEF2 support. |
| SofTech | Responsible for IDEF0 support. |

TASKS 4.3 - 4.9 (TEST BED)

| Subcontractors | Role |
|---|---|
| Boeing Military Aircraft Company (BMAC) | Responsible for consultation on applications of the technology and on IBM computer technology. |
| Computer Technology Associates (CTA) | Assisted in the areas of communications systems, system design and integration methodology, and design of the Network Transaction Manager. |
| Control Data Corporation (CDC) | Responsible for the Common Data Model (CDM) implementation and part of the CDM design (shared with DACOM). |
| D. Appleton Company (DACOM) | Responsible for the overall CDM Subsystem design integration and test plan, as well as part of the design of the CDM (shared with CDC). DACOM also developed the Integration Methodology and did the schema mappings for the Application Subsystems. |

| Subcontractors | Role |
|---|---|
| Digital Equipment Corporation (DEC) | Consulting and support of the performance testing and on DEC software and computer systems operation. |
| McDonnell Douglas Automation Company (McAuto) | Responsible for the support and enhancements to the Network Transaction Manager Subsystem during 1984/1985 period. |
| On-Line Software International (OSI) | Responsible for programming the Communications Subsystem on the IBM and for consulting on the IBM. |
| Rath and Strong Systems Products (RSSP) (In 1985 became McCormack & Dodge) | Responsible for assistance in the implementation and use of the MRP II package (PIOS) that they supplied. |
| SofTech, Inc. | Responsible for the design and implementation of the Network Transaction Manager (NTM) in 1981/1984 period. |
| Software Performance Engineering (SPE) | Responsible for directing the work on performance evaluation and analysis. |
| Structural Dynamics Research Corporation (SDRC) | Responsible for the User Interface and Virtual Terminal Interface Subsystems. |

Other prime contractors under other projects who have contributed to Test Bed Technology, their contributing activities and responsible projects are as follows:

| Contractors | ICAM Project | Contributing Activities |
|---|---|---|
| Boeing Military Aircraft Company (BMAC) | 1701, 2201, 2202 | Enhancements for IBM node use. Technology Transfer to Integrated Sheet Metal Center (ISMC). |

| Contractors | ICAM Project | Contributing Activities |
|---|---|---|
| Control Data Corporation (CDC) | 1502, 1701 | IISS enhancements to Common Data Model Processor (CDMP). |
| D. Appleton Company (DACOM) | 1502 | IISS enhancements to Integration Methodology. |
| General Electric | 1502 | Operation of the Test Bed and communications equipment. |
| Hughes Aircraft Company (HAC) | 1701 | Test Bed enhancements. |
| Structural Dynamics Research Corporation (SDRC) | 1502, 1701, 1703 | IISS enhancements to User Interface/Virtual Terminal Interface (UI/VTI). |
| Systran | 1502 | Test Bed enhancements. Operation of Test Bed. |

# TABLE OF CONTENTS

FIGURES

# SECTION 1

# SCOPE

## 1.1 Identification

This specification establishes the development, test and qualification requirements of a collection of form-based applications identified as the User Interface Services, known in this document as the UIS. The UIS is one configuration item of the Integrated Information Support System (IISS) User Interface (UI).

## 1.2 Functional Summary

The UIS consists of independent IISS programs which may be invoked from the IISS Function Screen. The UIS provides the following independent applications:

1) Define Application
2) Change Password
3) Message Management

# SECTION 2

# DOCUMENTS

## 2.1 Reference Documents

[1] ICAM Documentation Standards, 15 September 1983, IDS150120000C.

[2] Structural Dynamics Research Corporation, IISS Terminal Operator Guide, OM 620144000, November 1, 1985.

[3] Structural Dynamics Research Corportion, Forms Language Compiler Development Specification, DS 620144401B, 1 November 1985.

[4] Structural Dynamics Research Corporation, Forms Driven Form Editor Development Specification, DS 62014402B, 1 November 1985.

[5] Structural Dynamics Research Corporation, Report Writer Development Specification, DS 620144501B, 1 November 1985.

[6] Structural Dynamics Research Corporation, Rapid Application Generator Development Specification, DS 620144502 , 1 November 1985.

[7] Structural Dynamics Research Corporation, Text Editor Development Specification, DS 620144600B, 1 November 1985.

[8] Structural Dynamics Research Corporation, Application Interface Development Specification, DS 620144700 , 1 November 1985.

[9] Structural Dynamics Research Corporation, User Interface Services Development Specification, DS 620144100B, 1 November 1985.

[10] Structural Dynamics Research Corporation, Form Processor Development Specification, DS 620144200B, 1 November 1985.

[11] Structural Dynamics Research Corporation, Virtual Terminal Interface Development Specification, DS 620144300B, 1 November 1985.

[12] General Electric Co., System Design Specification, 7 February 1983.

## 2.2 Terms and Abbreviations

Application Interface: (AI), subset of the IISS User Interface that consists of the callable routines that are linked with applications that use the Form Processor or Virtual Terminal. The AI enables applications to be hosted on computers other than the host of the User Interface.

Application Process: (AP), a cohesive unit of software that can be initiated as a unit to perform some function or functions.

Form: structured view which may be imposed on windows or other forms. A form is composed of fields. These fields may be defined as forms, items, and windows.

Form Definition: (FD), forms definition language after compilation. It is read at runtime by the Form Processor.

Forms Definition Language: (FDL), the language in which electronic forms are defined.

Forms Driven Form Editor: (FDFE), subset of the FE which consists of a forms driven application used to create Form Definition files interactively.

Form Editor: (FE), subset of the IISS User Interface that is used to create definitions of forms. The FE consists of the Forms Driven Form Editor and the Forms Language Compiler.

Forms Language Compiler: (FLAN), subset of the FE that consists of a batch process that accepts a series of forms definition language statements and produces form definition files as output.

Form Processor: (FP), subset of the IISS User Interface that consists of a set of callable execution time routines available to an application program for form processing.

IISS Function Screen: the first screen that is displayed after logon. It allows the user to specify the function he wants to access and the device type and device name on which he is working.

Integrated Information Support System: (IISS), a test computing environment used to investigate, demonstrate and test the concepts of information management and information integration in the context of Aerospace Manufacturing. The IISS addresses the problems of integration of data resident on heterogeneous data bases supported by heterogeneous computers interconnected via a Local Area Network.

Presentation Schema: (PS), may be equivalent to a form. It is the view presented to the user of the application.

User Data: data which is either input by the user or output by the application programs to items.

User Interface: (UI), IISS subsystem that controls the user's terminal and interfaces with the rest of the system. The UI consists of two major subsystems: the User Interface Development System (UIDS) and the User Interface Management System (UIMS).

User Interface Development System: (UIDS), collection of IISS User Interface subsystems that are used by applications programmers as they develop IISS applications. The UIDS includes the Form Editor and the Application Generator.

User Interface Management System: (UIMS), the runtime UI. It consists of the Form Processor, Virtual Terminal, Application Interface, the User Interface Services and the Text Editor.

User Interface Services: (UIS), subset of the IISS User Interface that consists of a package of routines that aid users in controlling their environment. It includes message management, change password, and application definition services.

User Interface/Virtual Terminal Interface: (UI/VTI), another name for the User Interface.

Virtual Terminal: (VT), subset of the IISS User Interface that performs the interfacing between different terminals and the UI. This is done by defining a specific set of terminal features and protocols which must be supported by the UI software which constitutes the virtual terminal definition. Specific terminals are then mapped against the virtual terminal software by specific software modules written for each type of real terminal supported.

Window: dynamic area of a terminal screen on which predefined forms may be placed at run time.

Window Manager: a facility which allows the following to be manipulated: size and location of windows, the device on which an application is running, the position of a form within a window. It is part of the Form Processor.

# SECTION 3

# REQUIREMENTS

## 3.1 Computer Program Definition

The UIS is a collection of applications that use the Form Processor. Each application is individually accessed by filling its name in the function select form. Each application presents a form to the user which must be filled in with information for processing by that application. The application then communicates with the Form Processor through messages created and sent by the Application Interface and handled by the UIM (User Interface Monitor) of the Form Processor.

### 3.1.1 Interface Requirements

Each UIS application interfaces with the Network Transaction Manager (NTM) and the Forms Processor (FP) through the Application Interface (AI). The callable routines of the FP are used by the UIM to communicate with the user at the terminal. Communication with other IISS processes occurs through messages sent to and from the UIS applications via the NTM and the AI.

#### 3.1.1.1 Interface Block Diagram

Figure 3-1 is the interface block diagram that illustrates how the User Interface Services connect with the other elements of the system.

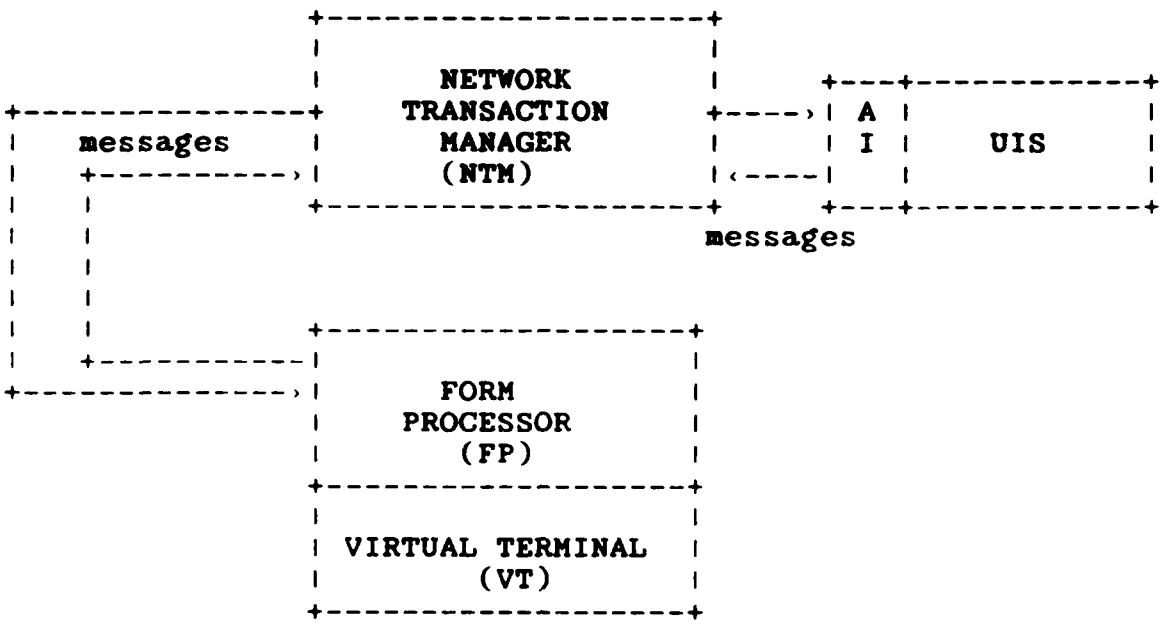



Figure 3-1 Interface Block Diagram

3.1.1.2 Detailed Interface Definition

The User Interface Services are applications which interface with the Form Processor through calls to the Application Interface (AI). The AI routines translte the applications reuest into the NTM message format when UIS applications are linked to support the NTM. UIS applications may also be linked stand-alone. Then they directly call the Form Processor routines.

The UIS applications as well as other IISS applications must also use the NTM initialization routine, INITAL and the NTM termination routine, TRMNAT when running with the NTM.

3.2 Detailed Function Requirements

The UIS applications are initiated through the IISS Function Screen menu presented to the user after logging on correctly to the IISS. Each application accepts input from and returns output to the user's terminal via application defined forms.

Specific functions within the User Interface Services permit the user to create, maintain and delete commands which are used to run application programs. Utility functions also exist for changing the user's password and for managing error messages for application development.

### 3.2.1 Input to UIS

The UIS are forms-driven applications. The forms for each application are described in the following sections.

### 3.2.2 Access to UIS

The User Interface Services like all IISS applications are initiated by the UIM of the Form Processor when the terminal user fills in the corresponding function on the IISS function screen.

### 3.2.3 UIS Functions

The User Interfce Services and their corresponding function names are:

1) Define Application - SDDEFINEAP
2) Change Password - SDPASSWORD
3) Message Management - SDMMZZZZZZ

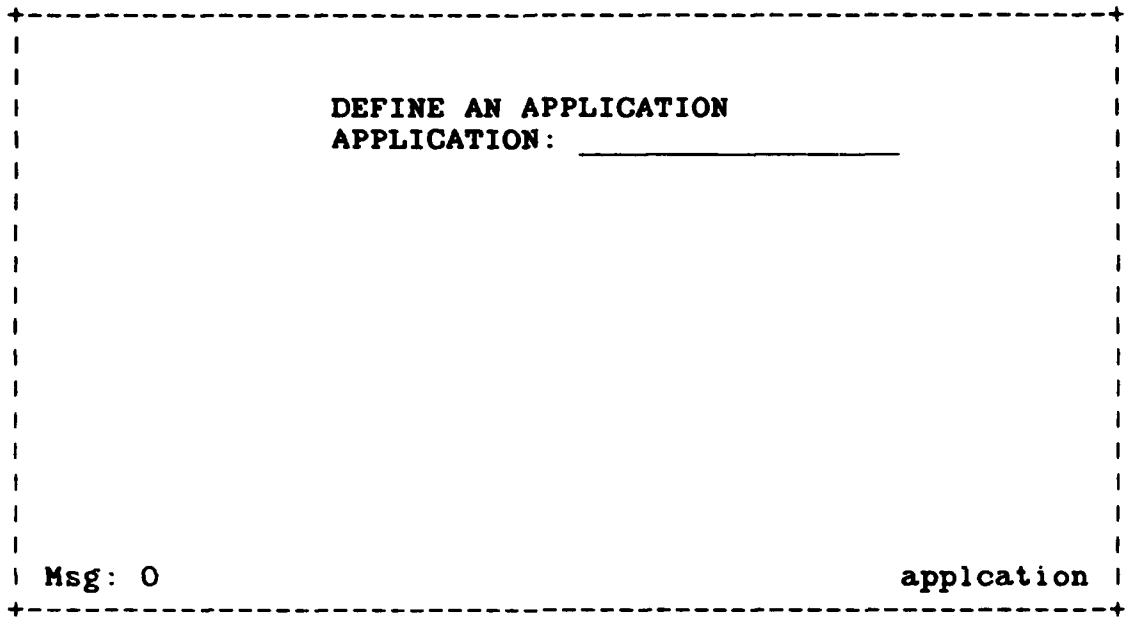
These UIS applications are individually described in the following subsections of this document.

#### 3.2.3.1 Define Application

Define Application is a UIS application which permits the user to define an application to the IISS. This UIS application allows the user to both define new applications and modify/delete existing applications. Section 3.2.3.1.1 describes the User Interface for this application.

##### 3.2.3.1.1 User Interface Description

When SDDEFINEAP is entered in the FUNCTION item, the following form is displayed.




APPLICATION is entered by the user, and is the name of of the application which will be used to run that application.

The form is transmitted by pressing the ENTER key once the application item is completed.

If the application entered on the DEFINE APPLICATION is new, the following form is displayed:

```
+-------------------------------------------------------------+
|                                                             |
|                  DEFINE AN APPLICATION                      |
|                  APPLICATION: _____________                 |
|                                                             |
|                                                             |
| DESCRIPTION: _____________ HOST: _____ CLUSTER: _____      |
|                                                             |
| COMMAND:                                                    |
|  _____________________________________________________     |
|  _____________________________________________________     |
|  _____________________________________________________     |
|  _____________________________________________________     |
|  _____________________________________________________     |
|  _____________________________________________________     |
|  _____________________________________________________     |
|  _____________________________________________________     |
|                                                             |
| Msg: 0                                          applcation  |
+-------------------------------------------------------------+
```

(1) APPLICATION identifies the application. The user entered it on the previous form.

(2) DESCRIPTION is a user-entered description of the application being defined. It is up to 20 alpha-numeric characters.

(3) HOST identifies the computer that hosts the application.

(4) CLUSTER identifies the IISS application cluster in which the application resides.

(5) COMMAND contains the message(s) that the Network Transaction Manager (NTM) sends to the application upon startup.

The form is transmitted by pressing the ENTER key when all items in the form are completed.

When the application entered on the DEFINE APPLICATION form already exists, The following form is displayed:

```
+-------------------------------------------------------------+
|                                                             |
|                  DEFINE AN APPLICATION                      |
|                  APPLICATION: _____________                 |
|                                                             |
| UPDATE: _                                                   |
| DELETE: _                                                   |
|                                                             |
|                                                             |
| DESCRIPTION: _____________ HOST: _____ CLUSTER: _____       |
|                                                             |
| COMMAND:                                                    |
| _____________________________________________________       |
| _____________________________________________ _________     |
| _____________________________________________ _________     |
| _____________________________________________________       |
| _____________________________________________________       |
| _____________________________________________________       |
| _____________________________________________________       |
| _____ _______________________________________________       |
|                                                             |
|                                                             |
| Msg: 0                                          applcation  |
+-------------------------------------------------------------+
```

(1) APPLICATION was entered by the user, and is the name of the application which will be used to run that application.

(2) UPDATE permits the user to update the data of the application.

(3) DELETE permits the user to delete the application.

(4) DESCRIPTION is a user-entered description of the application being defined. It is up to 20 alpha-numeric characters.

(5) HOST identifies the computer that hosts the application.

(6) CLUSTER identifies the IISS application cluster in which the application resides.

(7) COMMAND contains the message(s) that the Network Transaction Manager (NTM) sends to the application upon startup.

The form is transmitted by pressing the ENTER key when all items in the form are completed.

3.2.3.1.2 Database Description

A complete description of the database definitions are provided in section 3.5. This section describes what database tables and views are used by this application.

This UIS application accesses the APPL table and the ROLAPP table. It performs inserts, updates, and deletes on the APPL table and only performs deletes on the ROLAPP table. It does not use any database views.

3.2.3.1.3 Basic Architecture

The following figure shows the basic architecture of the Define Application application:

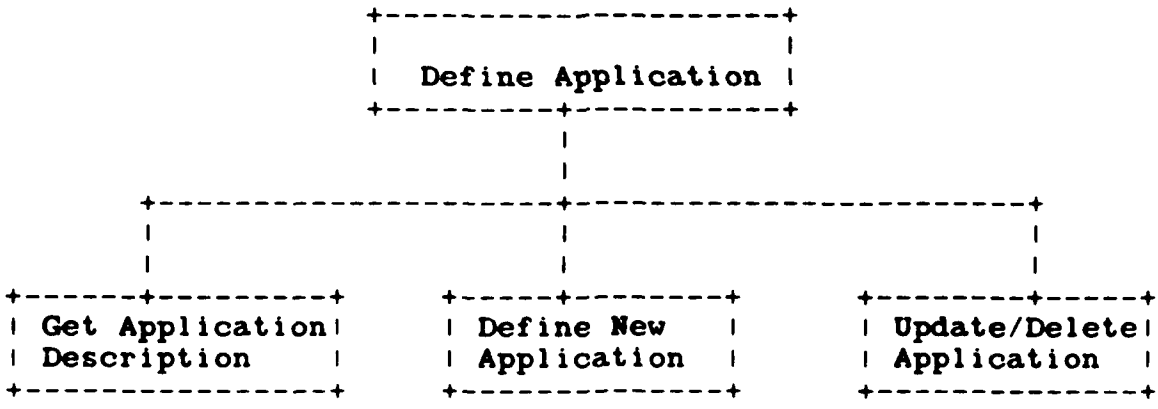



Figure 3-2 Basic Architecture of Define Application

The main function determines whether the entered application name is new or already exists. If it exists, then the application description is retrieved from the APPL table. The user may then modify or delete the application definition. Upon modification, the application's entry is modified in the APPL table. Upon deletion, the application's entry is deleted from the APPL table. Also, upon deletion, the application's entry in the ROLAPP table is also deleted. If the application is new, then the Define Application subfunction inserts the application into the APPL table.

### 3.2.3.2 Change Password

Change Password is a UIS application which permits the IISS user to change his/her password. Section 3.2.3.2.1 describes the User Interface for this UIS application.

#### 3.2.3.2.1 User Interface Description

To change the user password, the function "SDPASSWORD" is entered on the function select form. The following screen is then displayed.

```
+-------------------------------------------------------------+
|                                                             |
|                       CHANGE PASSWORD                       |
|                                                             |
| USER ID: _________                                          |
|                                                             |
| OLD PASSWORD: _________     NEW PASSWORD: __________        |
|                                                             |
|                             VERIFICATION: __________        |
|                                                             |
|                                                             |
|                                                             |
|                                                             |
|                                                             |
|                                                             |
|                                                             |
|                                                             |
|                                                             |
|                                                             |
|                                                             |
|                                                             |
| Msg: 0                                          applcation  |
+-------------------------------------------------------------+
```

USER ID is the user id whose password is to be changed. It must be entered.

OLD PASSWORD is the current password. It must be entered.

NEW PASSWORD is the password the user would like to have. It must be entered if the password is to be changed. This entry is not displayed on the screen for security reasons.

VERIFICATION verifies the new password choice. It must be entered. This entry is not displayed for security reasons.

3.2.3.2.2 Database Description

Section 3.5 shows a complete description of the data used in these UIS applications. The Change Password application only modifies the USER entity. The Define application adds; modifies, and deletes the application entity. It updates the PSWORD attribute of the user's record in the USER entity.

3.2.3.2.3 Basic Architecture

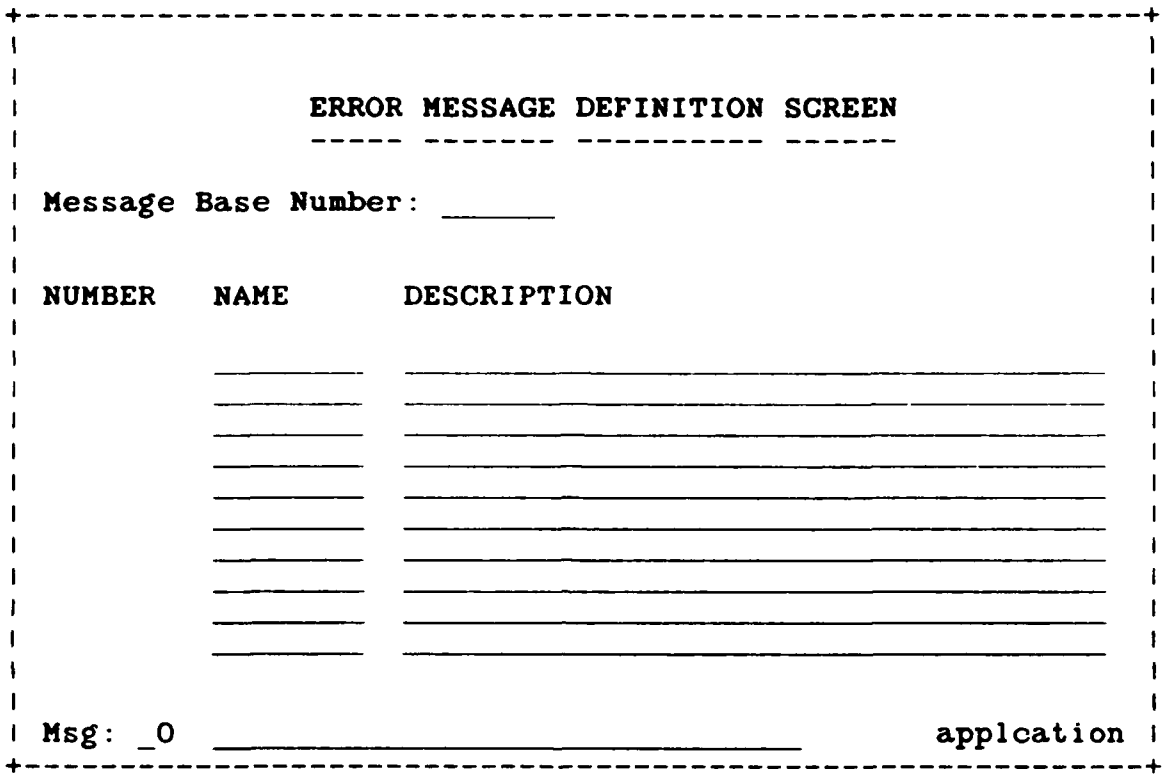
This application consists of two modules. The main module processes the application form. It retrieves the data from the form and verifies the input. Upon successful verification of password information, it calls the second module to update the user's entry in the UIUSER table. After update, it informs the user of the success of the update.

3.2.3.3 Message Management Application

Message Management (MM) is a predefined application of the UIS. It is used by programmers who are writing application software to run on the IISS. MM creates message code files to be used in conjunction with the Form Processor routine "PMSGLC".

3.2.3.3.1 User Interface Description

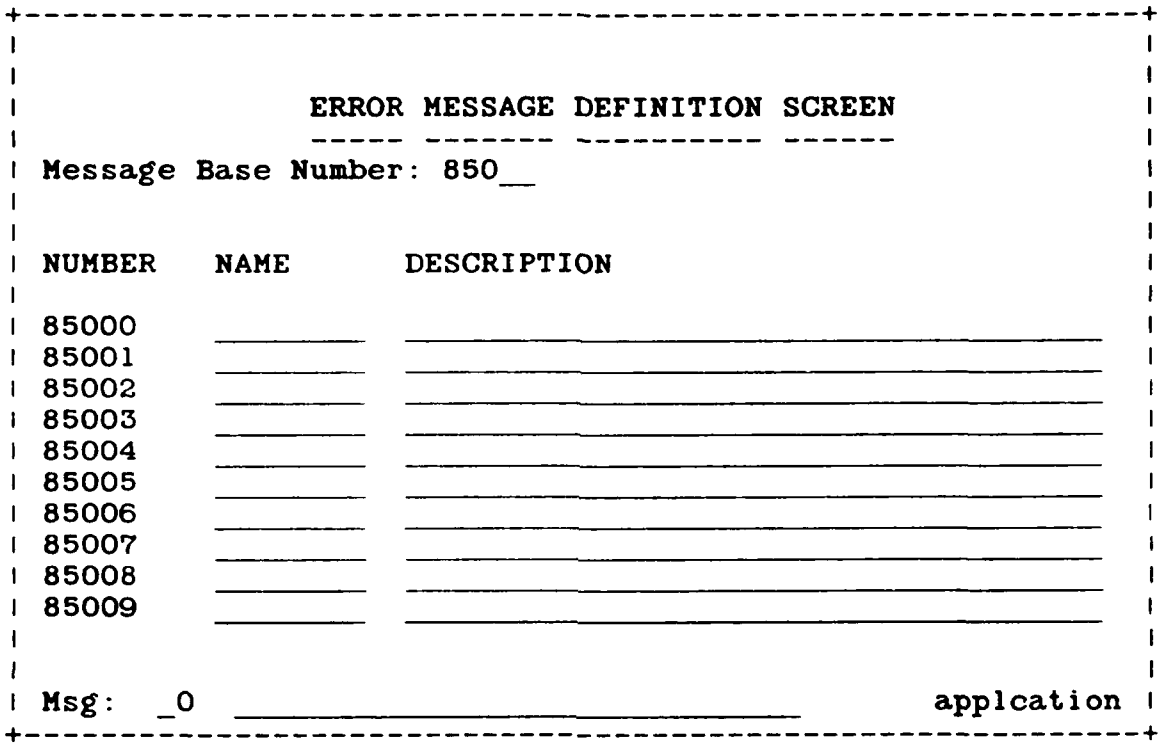
To run MM, "SDMMZZZZZZ" is entered as the FUNCTION on the function select form. The following screen is then displayed.




MESSAGE BASE NUMBER contains the first three (3) numbers of the message file. Any number from XXX00 to XXX99 is valid. XXX alone is also valid. If the message number entered already exists, the current names and messages are presented on the form. If a new number is entered, the form allows the user to enter the new names and messages.

3.2.3.3.2 Define New Error Messages

This is the form the user sees after entering a message number for which no messages have previously been defined. The message base of 850 is used as an example.




MESSAGE BASE NUMBER contains the first three (3) numbers of the message file. It was entered previously. If this number is changed, the DEFINE NEW MESSAGES form comes up if the number has no messages associated with it, or the UPDATE/DELETE MESSAGES form if the number has messages defined. Any number from XXX00 to XXX99 is valid.

NUMBER is the message number. This number can not be changed.

NAME is the name used in the application software to identify the message. It can be a maximum of 8 printable characters.

DESCRIPTION is the message which is displayed in the message line when the application invokes PMSGLC with the NUMBER. It can be a maximum of 60 characters.

If the message base already had existing messages defined for it, a form displaying these define messages would then be presented.

```
+-------------------------------------------------------------+
|                                                             |
|              ERROR MESSAGE DEFINITION SCREEN                |
|              ----- ------- ---------- ------                 |
|                                                             |
|  Message Base Number: 850__                                 |
|                                                             |
|                                                             |
| NUMBER   NAME      DESCRIPTION                              |
|                                                             |
| 85000    INVPOS    INVALID POSITION                         |
| 85001    IMPSEQ    IMPROPER SEQUENCE                        |
| 85002    SYNERR    SYNTAX ERROR                             |
| 85003    NVALCOM   NOT A VALID COMMAND                      |
| 85004    DUPFLD    DUPLICATE FIELD ENTRY,TRY AGAIN          |
| 85005    INVROW    INVALID ROW                              |
| 85006    _______   ___________________________________      |
| 85007    _______   ___________________________________      |
| 85008    _______   ___________________________________      |
| 85009    _______   ___________________________________      |
|                                                             |
|                                                             |
| Msg: _0  ___________________________________     applcation |
+-------------------------------------------------------------+
```

The NUMBER field may not be changed. The other items NAME and DESCRIPTION may be modified as the user desires. To delete a message definition, a user-established method such as entering "DELETED" in for the message DESCRIPTION would signal that this message NUMBER may be reused.

3.2.3.3.3 Database Description

In the VAX environment, the files that are updated are located in the directory pointed to by the logical IISSMLIB and have the format MSGXXX.MSG where XXX is the message base number. A new version of the MSGXXX.MSG file is created every time a new base number is chosen or an existing base number is updated. These message files should be purged so that several versions of the same file do not exist.

The record format of the MSGXXX.MSG file is the message number followed by the message name and then the message description. These files may then be inputted to a program called INCGEN. This program produces an include file in the language specified when the INCGEN program was executed. This include file may then be used in the application for which this message base was defined.

#### 3.2.3.3.4 Basic Architecture

This application consists of one module which processes the Message Management form. If the message base number previously exists, it reads the corresponding MSGXXX.MSG file to retrieve the current message definitions. Upon modification of the Message Definition information, this module updates the MSGXXX.MSG file.

## 3.3 Special Requirements

### 3.3.1 Programming Methods

UIS programming methods conform to the standards set forth by General Electric in the IISS Software Development Guidelines/Conventions document. Principles of structured design and programming are adherred to.

### 3.3.2 Expandability

To allow for flexibility and extensibility, the information used by the UIS as to what functions are available to a particular user as well as detailed information about each function should be stored in the Common Data Model (CDM) and accessed as required rather than being integral to the UIS. This release, however, uses the Oracle data base to store the information about the applications and users.

## 3.4 Human Performance

UIS is form-driven. Consideration was given to using menu-selection entry such as is used on many small computer systems available today. This approach was rejected. Instead entry of application names as the function selected was used due to the large number of applications which might exist. Cursoring through such a large list would take an unreasonable amount of time.

Menu-selection entry for function selection was also rejected for several reasons. First, it is conceivable that the list of functions may grow too large to be conveniently handled in one menu. Second, a hierarchy of menus is considered to be unwieldy and time-consuming for the current application. Third, no suitable way could be devised around the extra strokes of the ENTER key necessary for selecting a function which requires an application name. (The one exception where menu-selection entry is used is the index for selecting user-identifications pages; this entire index is a scrolling area.)

## 3.5 Data Base Requirements

### 3.5.1 Data Base Overview

The following IDEF1 model shows the entities and relationships of the data used by the UIS applications of Define Application and Change Password. A detailed mapping of this IDEF1 model to an ORACLE database is outlined in the Product Specification.

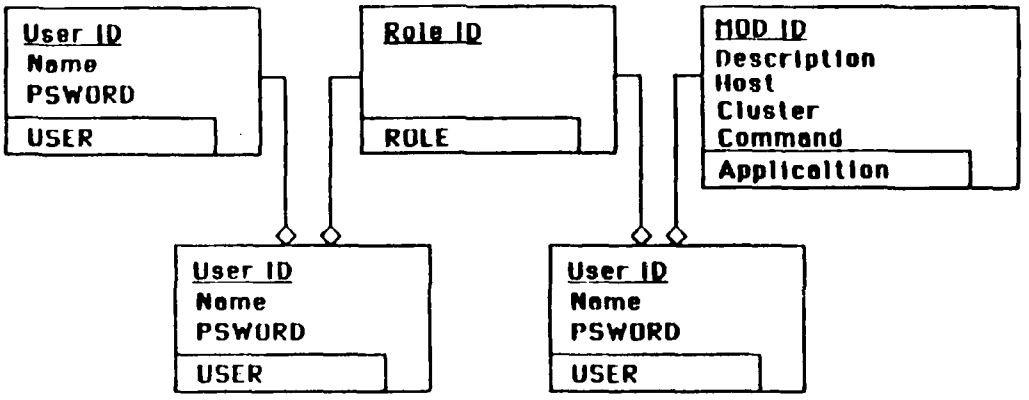



Figure 3-3 User/Role/Application Model

The following IDEF1 model shows the one entity of the Message Management application.

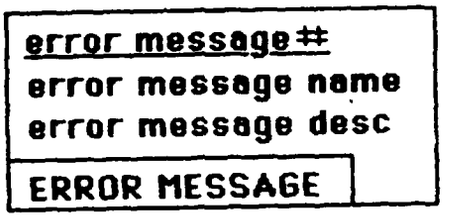



Figure 3-4 ERROR MESSAGE ENTITY

This entity is mapped to the operating system's file system.

# SECTION 4

# QUALITY ASSURANCE PROVISIONS

## 4.1 Introduction and Definition

"Testing" is a systematic process that may be preplanned and explicitly stated. Test techniques and procedures may be defined in advance and a sequence of test steps may be specified. "Debugging" is the process of isolation and correction of the cause of an error.

"Antibugging" is defined as the philosophy of writing programs in such a way as to make bugs less likely to occur and when they do occur, to make them more noticeable to the programmer and the user. In other words, as much error checking as is practical and possible in each routine should be performed. This approach was followed in the development of the UIS.

## 4.2 Computer Programming Test and Evaluation

The quality assurance provisions for test consist of the normal testing techniques that are accomplished during the construction process. They consist of design and code walk-throughs, unit testing, and integration testing. These tests are performed by the design team. Structured design, design walk-through and the incorporation of "antibugging" facilitate this testing by exposing and addressing problem areas before they become coded "bugs".

The integration testing entails use of the UIS applications on the VAX. These applications display forms, read input from forms, and display results.

Each application is tested separately. All testing except for integration with software belonging to other companies is done at SDRC on the VAX.

## SECTION 5

## PREPARATION FOR DELIVERY

The implementation site for the constructed software is the Integrated Support System (IISS) Test Bed site located at the General Electric Company, in Albany, New York. The software associated with each User Interface CPCI release is delivered on a media which is compatible with the IISS Test Bed. The release is clearly identified and includes instructions on procedures to be followed for installation of the release.

END

8-87

DTIC